Chapter 10
Model-Based Regression Testing:
Process, Challenges and Approaches

Qurat-ul-ann Farooq
Ilmenau University of Technology, Germany

Matthias Riebisch
Ilmenau University of Technology, Germany

ABSTRACT

Evolution is the consequence of the continuous changes a software system has to perform due to changes in requirements and various maintenance activities. Regression testing provides a means to assure the wanted properties of the system after the introduction of the changes; however, testing requires high effort. Model-based regression testing (MBRT) has the potential to perform test tasks with a much better efficiency. MBRT uses analysis and design models for identifying changes and their corresponding test cases to retest the system after modifications. MBRT promises reduction in cost and labour by selecting a subset of the test cases corresponding to the modifications. However, the identification of modifications in a system and the selection of corresponding test cases is challenging due to interdependencies among models.

This chapter aims to provide a detailed insight into MBRT, how it is related to the general software lifecycle, and what are the challenges involved. We evaluate the state of the art in MBRT with a detailed analysis of the strengths and weaknesses of the existing approaches. For the analysis, we develop a set of comprehensive analysis criteria based on the identified challenges. Furthermore, we demonstrate the applicability of MBRT by presenting our state-based MBRT approach as an example. The chapter targets researchers and practitioners who want to achieve a detailed comprehension of the field and want to know the strengths and weaknesses of the existing approaches in MBRT. This chapter also identifies the areas within MBRT which require further attention by the researchers.

DOI: 10.4018/978-1-61350-438-3.ch010

To improve is to change; to be perfect is to change often (Winston Churchill)

INTRODUCTION

Evolution is inherent to the software systems. Due to the growing size and complexity of modern systems, the evolving nature of a system can cause adverse side effects and even system failures. Besides many other measures to prevent these unintended effects of evolution, it is essential to test a system after modifications; often referred to as *Regression Testing*. Regression testing is performed during the software maintenance phase and during various maintenance activities including *corrective*, *perfective* and *adaptive* maintenance (Wu & Offutt, 2003).

When a software system is modified, repeating the entire testing activity is a very costly task. A large system may have a huge number of test cases and test-execution requirements. Executing all these test cases is generally not a economically feasible option. Hence, it is necessary for regression testing to select a subset of the test cases corresponding to modifications. This is known as the selective strategy for regression testing and is a more feasible solution in terms of cost and time (Binder, 1999).

Another important issue during regression testing is scalability. Conventional code-based regression testing approaches fail to deal with the huge size of modern software systems. Model-based regression testing is a potential solution to this problem, because it offers several advantages compared to the conventional code-based regression testing approaches. This includes *better scalability*, *better complexity management* and *better comprehension* of the system, the relevant test suites and test cases. In model-based testing, the testing activity can be started in early phases of software development allowing *early regression planning and estimation* (Briand, Labiche, & He, 2009). This results in *effort reduction* in terms of time and labour. Furthermore, traceability *maintenance* between test cases and models is relatively easier to accomplish as compared to the code-based approaches (Briand, Labiche, & He, 2009). Due to the use of models as primary artefact in the MBRT, static and dynamic *interactions are more visible* in design models and no static and dynamic analysis is required to determine the dynamic bindings as in code based approaches. Finally, *portability* and *platform independence* is a major benefit for evolving systems to adopt the rapid changes in technology and operational environment.

Besides all these benefits, there are some limitations of model-based regression testing as well. One of the major limitations is the potential impact of incomplete and outdated design models on the creation of effective regression test suites and plans. Moreover, since the test cases are generated from the design models, they are more abstract than test cases generated from code. This abstraction, sometimes make the test execution more difficult as the test cases should be adapted according to the implementation environment. However, considering the benefits of model-based regression testing, these limitations can be somehow compromised.

In this chapter, we try to not only give a broad overview of the area but also discuss all the main steps and key challenges involved in model-based regression testing. We discuss the role and place of MBRT in the classical software development lifecycle and identify the major steps involved in the MBRT phase. We also identify and discuss the challenges associated with model-driven regression testing approaches which are relatively novice approaches and are influenced by the concepts of model-driven architecture (MDA, 2011). We present our state-based regression testing approach with a demonstrating example to apply the regression testing steps identified previously and give a detailed insight of how practical regression testing can be performed to the reader. As the major contribution of the chapter, we provide a comparative analysis of the existing